



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : H04N 7/36, 7/26	A1	(11) International Publication Number: WO 00/18134 (43) International Publication Date: 30 March 2000 (30.03.00)
(21) International Application Number: PCT/US99/21830 (22) International Filing Date: 20 September 1999 (20.09.99) (30) Priority Data: 60/100,939 18 September 1998 (18.09.98) US 09/255,946 23 February 1999 (23.02.99) US (71) Applicant: SARNOFF CORPORATION [US/US]; CN5300, 201 Washington Road, Princeton, NJ 08543-5300 (US). (72) Inventors: SETHURAMAN, Sriram; Apartment #28-B, Kensington Arms, Hightstown, NJ 08520 (US). KRISHNA-MURTHY, Ravi; 5614 Hunters Glen Drive, Plainsboro, NJ 08536 (US). (74) Agents: BURKE, William, J. et al.; Sarnoff Corporation, CN5300, 201 Washington Road, Princeton, NJ 08543-5300 (US).		(81) Designated States: BR, CA, CN, IN, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report.</i>
(54) Title: FRAME SKIPPING WITHOUT HAVING TO PERFORM MOTION ESTIMATION <div style="margin-left: 100px;"> 1 $H = \text{abs}((MAD(I) - MAD(I-1)) * 2 / (MAD(I) + MAD(I-1)))$ 2 If ($H < T1$) 3 $Se(I) = S(I-1)$ 4 Elseif ($H < T2$) 5 $Se(I) = S(I-1) + (k * MAD(I) - S(I-1)) * (H - T1) / (T2 - T1)$ 6 Else 7 $Se(I) = k * MAD(I)$ </div> (57) Abstract <p>A raw distortion measure (e.g., <i>MAD</i>), relative to a reference frame, is generated for the current image in a video stream. The raw distortion measure is then used to generate an estimate (e.g., <i>Se</i>) of a motion-compensated distortion measure (e.g., <i>S</i>) for the current image relative to the reference image without having to perform motion estimation on the current image. The estimate of the motion-compensated distortion measure is then used to determine whether or how to encode the current image, and a compressed video bitstream is generated for the sequence of video images based, in part, on that determination. The present invention enables a video coder to determine when to skip frames without first having to perform computationally expensive motion estimation and motion compensation processing, which computation load would otherwise be wasted for frames that are skipped.</p>		

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Larvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

FRAME SKIPPING WITHOUT HAVING TO PERFORM MOTION ESTIMATION

BACKGROUND OF THE INVENTIONField of the Invention

5 The present invention relates to image processing, and, in particular, to video compression.

Cross-Reference to Related Applications

 This application claims the benefit of the filing date of U.S. provisional application no. 60/100,939, filed on 09/18/98 as attorney docket no. SAR 12728PROV.

10

Description of the Related Art

 In video compression processing, it is known to encode images using motion-compensated inter-frame differencing in which blocks of image data are encoded based on the pixel-to-pixel differences between each block in an image currently being encoded and a selected block in a reference
15 image. The process of selecting a block in the reference image for a particular block in the current image is called motion estimation. The goal of motion estimation is to find a block in the reference image that closely matches the block in the current image such that the magnitudes of the pixel-to-pixel differences between those two blocks are small, thereby enabling the block in the current image to be encoded in the resulting compressed bitstream using a relatively small number of bits.

20

 In a typical motion estimation algorithm, a block in the current image is compared with different blocks of the same size and shape within a defined search region in the reference image. The search region is typically defined based on the corresponding location of the block in the current image with allowance for inter-frame motion by a specified number of pixels (e.g., 8) in each direction. Each comparison involves the computation of a mathematical distortion measure that quantifies the
25 differences between the two blocks of image data. One typical distortion measure is the sum of absolute differences (SAD) which corresponds to the sum of the absolute values of the corresponding pixel-to-pixel differences between the two blocks, although other distortion measures may also be used.

 There are a number of methods for identifying the block of reference image data that "best" matches the block of current image data. In a "brute force" exhaustive approach, each possible
30 comparison over the search region is performed and the best match is identified based on the lowest distortion value. In order to reduce the computational load, alternative schemes, such as log-based or layered schemes, are often implemented in which only a subset of the possible comparisons are performed. In either case, the result is the selection of a block of reference image data as the block that "best" matches the block of current image data. This selected block of reference image data is referred
35 to as the "best integer-pixel location," because the distance between that block and the corresponding location of the block of current image data may be represented by a motion vector having X

(horizontal) and Y (vertical) components that are both integers representing displacements in integer numbers of pixels. The process of selecting the best integer-pixel location is referred to as full-pixel or integer-pixel motion estimation.

5 In order to improve the overall encoding scheme even further, half-pixel motion estimation may be performed. In half-pixel motion estimation, after performing integer-pixel motion estimation to select the best integer-pixel location, the block of current image data is compared to reference image data corresponding to different half-pixel locations surrounding the best integer-pixel location, where the comparison for each half-pixel location is based on interpolated reference image data.

10 Even though some of these motion estimation techniques require fewer computations than other techniques, they all require a significant computational effort.

The primary goal in video compression processing is to reduce the number of bits used to represent sequences of video images while still maintaining an acceptable level of image quality during playback of the resulting compressed video bitstream. Another goal in many video compression applications is to maintain a relatively uniform bit rate, for example, to satisfy transmission bandwidth and/or playback processing constraints.

15 Video compression processing often involves the tradeoff between bit rate and playback quality. This tradeoff typically involves reducing the average number of bits per image in the original video sequence by selectively decreasing the playback quality in each image that is encoded into the compressed video bitstream. Alternatively or in addition, the tradeoff can involve skipping certain images in the original video sequence, thereby encoding only a subset of those original images into the resulting compressed video bitstream.

20 Conventional video compression algorithms dictate a regular pattern of image skipping, e.g., skip every other image in the original video sequence. In addition, a video encoder may be able to skip additional images adaptively as needed to satisfy bit rate requirements. The decision to skip an additional image is typically based on a distortion measure (e.g., SAD) of the motion-compensated interframe differences and only after motion estimation has been performed for the particular image.

25 When the decision is made not to skip the current frame, the motion-compensated interframe differences derived from the motion estimation processing are then used to further encode the image data (e.g., depending on the exact video compression algorithm, using such techniques as discrete cosine transform (DCT) processing, quantization, run-length encoding, and variable-length encoding). On the other hand, when the decision is made to skip the current frame, the motion-compensated interframe differences are no longer needed, and processing continues to the next image in the video sequence.

35

SUMMARY OF THE INVENTION

The present invention is directed to a technique for generating an estimate of a motion-compensated distortion measure for a particular image in a video sequence without actually having to perform motion estimation for that image. In preferred embodiments, the estimated distortion measure can be used during video encoding to determine whether to skip the image without first having to perform motion estimation. When the decision is made to skip the image, motion estimation processing is avoided and the computational load of the video compression processing is accordingly reduced. When the decision is made to encode the image, motion estimation processing can then be implemented, as needed, to generate motion-compensated interframe differences for subsequent compression processing. Under such a video compression scheme, motion estimation processing is implemented only when the resulting interframe differences will be needed to encode the corresponding image.

According to one embodiment, the present invention is a method for processing a sequence of video images, comprising the steps of (a) generating a raw distortion measure for a current image in the sequence relative to a reference image; (b) using the raw distortion measure to generate an estimate of a motion-compensated distortion measure for the current image relative to the reference image without having to perform motion estimation on the current image; (c) determining whether or how to encode the current image based on the estimate of the motion-compensated distortion measure; and (d) generating a compressed video bitstream for the sequence of video images based on the determination of step (c).

BRIEF DESCRIPTION OF THE DRAWINGS

Other aspects, features, and advantages of the present invention will become more fully apparent from the following detailed description, the appended claims, and the accompanying drawings in which:

Fig. 1 shows pseudocode for an algorithm for generating a raw (i.e., non-motion-compensated) distortion measure for an image, according to one embodiment of the present invention;

Fig. 2 shows pseudocode for an algorithm for estimating a motion-compensated distortion measure for an image, according to one embodiment of the present invention; and

Figs. 3A-3C provide pseudocode for an algorithm for determining what frames to code and how to code them, according to one embodiment of the present invention.

DETAILED DESCRIPTION

Generating a Raw Distortion Measure for a Current Image

Fig. 1 shows pseudocode for an algorithm for generating a raw (i.e., non-motion-compensated) distortion measure for an image, according to one embodiment of the present invention. The particular raw distortion measure generated using the algorithm of Fig. 1 is a mean absolute difference *MAD*.

The algorithm in Fig. 1 can be interpreted as applying to gray-scale images in which each pixel is represented by a single multi-bit intensity value. It will be understood that the algorithm can be easily extended to color images in which each pixel is represented by two or more different multi-bit components (e.g., red, green, and blue components in an RGB format or an intensity (Y) and two color (U and V) components in a YUV format).

The algorithm of Fig. 1 distinguishes two different types of pixels in the current image: Type I being those pixels having an intensity value sufficiently similar to the corresponding pixel value in the reference image and Type II being those pixels having a pixel value sufficiently different from that of the corresponding pixel in the reference image. In this algorithm, the "corresponding" pixel is the pixel in the reference image having the same location (i.e., same row and column) as a pixel in the current image.

When there is no motion in the imagery depicted, between a portion of the reference image and the corresponding portion of the current image, then the pixels in that portion of the current image will typically be characterized as being of Type I. Similarly, when there is motion between relatively spatially uniform portions (i.e., portions in which the pixels have roughly the same value), those pixels will also typically be characterized as being of Type I. If, however, there is motion between spatially non-uniform portions, the absolute differences between the pixels in the current image and the corresponding pixels in the reference image will be relatively large and most of those current-image pixels will typically be characterized as being of Type II.

The variables $n1$ and $n2$ are counters for these two different types of pixels, respectively, and the variables $dist1$ and $dist2$ are intermediate distortion measures for these two different types of pixels, respectively. For each new image, these four variables are initialized to zero at Lines 1-2 in Fig. 1.

For each pixel in the current image (Line 3), the absolute difference ad between the current pixel value and the corresponding pixel value in the reference frame is generated (Line 4). If ad is less than a specified threshold value $thresh$, then the current pixel is determined to be of Type I, and $dist1$ and $n1$ are incremented by ad and 1, respectively (Line 5). Otherwise, the current pixel is determined to be of Type II, and $dist2$ and $n2$ are incremented by ad and 1, respectively (Line 6). In order to pick up significant edges, a typical threshold value for the parameter $thresh$ is about 20. The intermediate distortion measures $dist1$ and $dist2$ are then normalized in Lines 8 and 9, respectively.

In the case of the video-conferencing paradigm of a talking head in front of a uniform background (e.g., a uniformly painted wall), relative movement of the person's head from frame to frame (e.g., a side-to-side motion) will result in some portions of the wall being newly covered by pixels corresponding to the head and other portions of the wall that were previously occluded by the head being newly exposed. Such a situation will result in two different significant edges in the raw interframe differences: one edge corresponding to those portions of the background newly covered by

the head and a second edge corresponding to those portions of the background newly uncovered by the head. These two edges are referred to as double-image effects.

The raw distortion measure *MAD*, generated using the expression in Line 10, is a mean absolute difference that is corrected for double-image effects. In order to avoid double counting of significant edges, a typical value for the parameter *factor* is 0.5. The term $(dist1 * n2 * (1 - factor))$ corrects for double-image effects by treating pixels removed from Type II as Type I pixels so that the average distortion level in similar areas is added back. The distortion *dist1* of Type I pixels is considered as an estimate for the residual and coding noise. It is assumed that this cannot be removed by motion compensation. The Type II pixels occupy roughly twice the area as compared to the "perfectly" motion-compensated images, and the term *factor* reflects this, and is nominally chosen as 0.5. The term *factor* is allowed to vary, since motion compensation is typically not perfect. It is assumed that the unoccluded region can be motion compensated; however, the fraction of pixels $(n2 * (1 - factor))$ is expected to have a residual plus coding noise similar to Type I pixels. Hence, the term $dist1 * n2 * (1 - factor)$ is used as an estimate for distortion of these unoccluded Type II pixels.

Generating an Estimated Motion-Compensated Distortion Measure from the Raw Distortion Measure

Fig. 2 shows pseudocode for an algorithm for estimating a motion-compensated distortion measure for an image, according to one embodiment of the present invention. The particular distortion measure estimated using the algorithm of Fig. 2 is the motion-compensated mean absolute difference *S*. The algorithm of Fig. 2 derives an estimate *Se* for the distortion measure *S* from the raw distortion measure *MAD* derived using the algorithm of Fig. 1. This estimated distortion measure *Se* can be used to determine whether to skip images during video encoding without having to perform motion estimation processing for each image.

According to the algorithm of Fig. 2, the raw distortion measure *MAD(I)* for the current frame and the raw distortion measure *MAD(I-1)* for the previous frame are used to determine a measure *H* of the percentage change in *MAD* from the previous frame to the current frame (Line 1 of Fig. 2). Other suitable expressions characterizing the change in the raw distortion measure *MAD* from the previous frame to the current frame could also conceivably be used.

If the percentage change *H* is less than a first threshold value *T1* (Line 2), then the estimated distortion measure *Se(I)* for the current frame is assumed to be the same as the actual motion-compensated distortion measure *S(I-1)* for the previous frame (Line 3). Otherwise, if the percentage change *H* is less than a second threshold value *T2* (Line 4) (where *T2* is greater than *T1*), then the estimated distortion measure *Se(I)* for the current frame is determined using the expression in Line 5, where the factor *k* is a parameter preferably specified between 0 and 1. Otherwise, the percentage change *H* is greater than the second threshold value *T2* (Line 6) and the estimated distortion measure

$Se(I)$ for the current frame is determined using the expression in Line 7. Typical values for $T1$ and $T2$ are 0.1 and 0.5, respectively.

The motivation behind the processing of Fig. 2 is as follows. The raw distortion measure $MAD(I)$ is a measure of the non-motion-compensated pixel differences between the current frame and its reference frame. Similarly, the raw distortion measure $MAD(I-1)$ is a measure of the raw pixel differences between the previous frame and its reference frame, which may be the same as or different from the reference frame for the current frame. The percentage change H is a measure of the relative change between the two raw distortion measures $MAD(I)$ and $MAD(I-1)$, which are themselves measures of rates of change between those images and their corresponding reference images.

Motion compensation does a fairly good job predicting image data when there is little or no change in distortion from frame to frame. As such, when the percentage change H is small (e.g., $H < T1$), the actual motion-compensated distortion measure $S(I-1)$ for the previous frame will be a good estimate $Se(I)$ of the motion-compensated distortion measure $S(I)$ for the current frame, as in Line 3 of Fig. 2.

However, when the distortion from frame to frame is changing (e.g., during a scene changes or other non-uniform changes in imagery), motion compensation will not do as good a job predicting the image data. In these situations, the actual motion-compensated distortion measure $S(I-1)$ for the previous frame will not necessarily be a good indication of the actual motion-compensated distortion measure $S(I)$ for the current frame. Thus, when the percentage change H is large (e.g., $H > T2$), it may be safer to estimate the actual motion-compensated distortion measure $S(I)$ for the current frame from the raw distortion measure $MAD(I)$ for the current frame, as in the expression in Line 7 of Fig. 2. Selecting the factor k to be between 0 to 1 (e.g., preferably 0.8) assumes that motion-compensation will typically reduce the distortion measure by some specified degree.

The expression in Line 5 of Fig. 2 provides a linear interpolation between these two "extreme" cases for situations where the percentage change H is neither small nor large (e.g., $T1 < H < T2$). As such, the algorithm of Fig. 2 provides a piecewise-linear, continuous relationship between the raw distortion measure MAD and the estimated motion-compensated distortion measure Se for all values of MAD . Experiment results confirm that the algorithms of Figs. A and B provide a reliable estimate Se of the actual motion-compensated distortion measure S , where the estimated distortion measure Se is almost always within 20% of the actual distortion measure S , and usually within 10-15%.

Determining Whether To Skip the Current Image Using the Estimated Distortion Measure

The estimated distortion measure Se generated using the algorithms of Figs. A and B can be used to determine whether to skip the current image, that is, whether to avoid encoding the current image into the compressed video bitstream during video encoding processing. In one embodiment of the present invention, an adaptive frame-skipping scheme enables a video coder to maintain control

over the transmitted frame rate and the quality of the reference frames. In cases of high motion, this ensures a graceful degradation in frame quality and the frame rate.

The coder can be in one of two states: steady state or transient state. In the steady state, all attempts are made to meet a specified frame rate, and, if this is not possible, an attempt is made to maintain a certain minimum frame rate. When it becomes impossible to maintain even the minimum frame rate, the coder switches into a transient state, where large frame skips are allowed until the buffer level depletes and the next frame can be transmitted. In addition to the start of transmission, the transient state typically occurs during scene changes and sudden large motions. It is desirable for the coder to go from the transient state to the steady state in a relatively short period of time.

Depending on the video compression algorithm, images may be designated as the following different types of frames for compression processing:

- o An intra (I) frame which is encoded using only intra-frame compression techniques,
- o A predicted (P) frame which is encoded using inter-frame compression techniques based on a previous I or P frame, and which can itself be used as a reference frame to encode one or more other frames,
- o A bi-directional (B) frame which is encoded using bi-directional inter-frame compression techniques based on a previous I or P frame and a subsequent I or P frame, and which cannot be used to encode another frame, and
- o A PB frame which corresponds to two images -- a P frame and a temporally preceding B frame -- that are encoded as a single frame with a single set of overhead data (as in the H.263 video compression algorithm).

According to one embodiment of the present invention, in the transient state, only I and P frames are allowed, while, in the steady state, B frames (H.263+, MPEG) and PB frames (H.263) are also allowed. In the steady state, the B and PB frames are used for two purposes in two different situations. First, when motion is large, B frames are used to increase the frame rate to acceptable levels. Second, when motion is small, using B and/or PB frames enables achievement of higher compression efficiency. The system is designed for applications where control over the rate and quality of reference frames is required. The parameters that are adjusted include the rate for the frame, the acceptable distortion level in the frame, and the frame-rate. An attempt is made to maintain these parameters by performing an intelligent mode decision as to when to encode B or PB frames and by intelligently skipping frames, when warranted.

These decisions are based on estimates of rate and distortion parameters that are measured as the frames are read into the frame buffer, which fit in very well with the H.263+ Video Codec Near-Term Model 8 (TMN 8, Study Group 16, ITU-T, Document Q15-A-59, Release 0, June 1997), and certain rate control schemes that can be used for MPEG and H.263. The strategy also ensures that the minimal amount of storage is used for the incoming frames that are encoded. Other strategies are

possible that use more storage but enable maintenance of a better control on frame rate and reference frame quality. The present strategy also ensures that computational overhead for extra motion estimation is minimal. If additional computational power is available for motion estimation, the performance of the algorithm can be improved further.

- 5 This strategy is based on a quadratic rate distortion model that relates the rate for encoding the frame and the SAD (sum of absolute differences) after motion compensation. This model is shown in Equation (1) as follows:

$$(R-H) / S = X1/Q + X2 / (Q**2), \quad (1)$$

where:

- 10 **R:** Number of bits needed to encode the current frame as a P frame. The same model can be applied for a B frame except with a quantizer that is typically higher than that of a corresponding P frame;
- H:** Number of bits needed to encode the overhead (i.e., header and motion information);
- S:** Motion-compensated interframe SAD over the current frame;
- 15 **Q:** Average quantizer step size over the previous frame; and
- X1, X2:** Parameters of quadratic model, which are recursively updated from frame to frame.

- Since it is desirable to avoid estimating motion for frames that are not going to be encoded, the estimate S_e , generated using the algorithms of Figs. A and B without have to perform motion
- 20 estimation, is preferably used in Equation (1) for the motion-compensated distortion measure S . Although the model has been described using the sum of absolute differences as the cost function, the present invention can be implemented using other suitable cost functions.

- Consider, as an example, a sequence of three frames A, C, and E, where temporally Frame A is the first of the three frames and Frame E is the last of the three frames. The following discussion is for
- 25 PB frames or for coding schemes having at most one B frame between reference frames. Generalizations to coders with more than one B frame in between reference frames will be described later. Assuming that Frame A is encoded as a reference frame (i.e., as either an I or a P frame), the decision needs to be made as to how to encode Frames C and E, if at all. The following four choices are possible:

- 30 (1) Encode Frame C as a B frame and encode Frame E as a reference frame;
- (2) Encode Frames C and E together as a PB frame;
- (3) Encode Frame C as a reference frame and restart process to determine how to encode Frame E;
- and
- (4) Skip Frame C and encode Frame E as a reference frame.
- 35 If possible, it is desirable to encode Frames C and E together as a PB frame. When motion is large and the buffer occupancy is not too high, Frame C may need to be encoded as a reference frame, in which

case, the process is restarted to determine how to encode Frame E. When motion is large and the buffer occupancy is too high, Frame C may need to be skipped, in which case, Frame E will be encoded as a reference frame. The subsequent discussion assumes that the time reference is at Frame A.

5

Notation

The following notation is used in the algorithm described in detail later in this specification.

MAD Raw distortion measure for the current frame, where the distortion measure is based on the mean absolute difference.

10 *S* Actual motion-compensated distortion measure for the current frame, where the distortion measure is based on the mean absolute difference.

Se Estimate of the actual motion-compensated distortion measure *S* for the current frame, where the distortion measure is based on the raw mean absolute difference *MAD*.

R Number of bits needed to encode the current frame, generated according to Equation (1) using
15 either the estimated distortion measure *Se* or the actual distortion measure *S*.

H Overhead bits (e.g., for motion vectors) other than bits used to transmit residuals for the current frame. If this information is unavailable, *H* is assumed to be zero.

Rp Bits output to the channel in one picture interval in the constant bit rate (CBR) case.

smin Smallest skip desired for encoding the next frame (e.g., 1 / average target frame rate).

20 *smax* Largest skip allowed between frames at steady state.

skip Pointer corresponding to the number of frames to skip from the previously encoded frame.

Bframeskip Pointer corresponding to frame stored as a potential B frame.

Bmax Total size of the buffer.

B Buffer occupancy at frame *skip* before encoding frame *skip*. For a constant bit rate channel,
25 $B = B_p - (R_p * skip)$, where *Bp* is the buffer occupancy after encoding the previous frame.

The algorithm relies on the following flags:

PCFDI: Indicates whether there is enough room in the buffer to transmit the current frame as a P
30 frame, where that determination is made without first performing motion estimation for the current frame. In one embodiment, if $(R(Se) + B < x * B_{max})$, where *R* is generated using Equation (1) based on the estimated distortion measure *Se*, then there is room in the buffer and *PCFDI*=1. Otherwise, there is not enough room in the buffer and *PCFDI*=0. In one implementation, $x=80\%$, although the tightness of the constraint can be varied by changing the value of *x*.

35

PCI: Similar to *PCFD1*, except that *R* is generated using Equation (1) after performing motion estimation and based on the actual distortion measure *S*.

5 *PCFD2*: Indicates whether motion in the current frame relative to its reference frame is "large," where that determination is made without first performing motion estimation for the current frame. In this case, the magnitude of motion is based on the raw distortion measure *MAD*. If *MAD* is greater than a specified threshold level, then motion is said to be large and *PCFD2*=1. Otherwise, motion is not large and *PCFD2*=0.

10 *PC2*: Similar to *PCFD2*, except that the determination is made after motion estimation, e.g., by comparing the average motion vector magnitude to a specified threshold level.

15 *PBCFD*: Indicates whether the current frame and a previous frame stored as a potential B frame can be coded together as a PB frame, where that determination is made without first performing motion estimation for the current frame. In one embodiment, if $(R(S_e) + (\text{bits to encode B frame}) + B < x \cdot B_{\text{max}})$, then the two frames can be encoded together as a PB frame and *PBCFD*=1. Otherwise, they cannot and *PBCFD*=0.

20 *Pmeet* Indicates whether a previously stored frame can be transmitted as a P frame. If so, *Pmeet*=1.

25 Figs. 3A-3C provide pseudocode for an algorithm for determining what frames to code and how to code them, according to one embodiment of the present invention. The algorithm contains seven routines: *START*, *LOOP1-LOOP5*, and *TRANSIENT*. *START* is called during steady-state processing after coding a reference frame. *TRANSIENT* is called during transient processing. As described earlier, in the steady state, all attempts are made to meet the preset specified frame rate, and, if this is not possible, an attempt is made to maintain a certain minimum frame rate. When it becomes impossible to maintain even the minimum frame rate, the coder switches into the transient state, where large frame skips are allowed until the buffer level depletes and the next frame can be transmitted. The transient state typically occurs at the start of the transmission, during scene changes, and during sudden large motions.

START Routine

35 The processing of the *START* routine begins at Line A1 in Fig. 3A with the initialization of the current frame pointer *skip* to the minimum skip value *smin*. For example, in one embodiment, the smallest frame skip value may be 2, corresponding to a coding scheme in which an attempt is made to

encode every other image in the original video sequence. The raw distortion measure *MAD* is computed for the current frame *skip* using the algorithm of Fig. 1. After using the algorithm of Fig. 2 to generate the estimated motion-compensated distortion measure *Se* from the raw distortion measure *MAD*, Equation (1) is then evaluated using *Se* to estimate *R*, the number of bits needed to encode the current frame as a P frame. If encoding the current frame as a P frame does not make the buffer too full, then the flag *PCFDI* is set to 1 (i.e., true). Otherwise, *PCFDI* is set to 0 (i.e., false).

If *PCFDI* is true (Line A2) indicating that the current frame can be transmitted as a P frame, then motion estimation is performed for the current frame, the actual motion-compensated distortion measure *S* is calculated, the number of bit *R* is reevaluated using *S* in Equation (1) instead of *Se*, and the values for flags *PC1* and *PC2* are determined (Line A3). The flag *PC1* indicates the impact to the buffer from encoding the current frame *skip* as a P frame based on the motion-compensated distortion measure *S*. Like *PCFDI*, *PC1* is set to 1 if frame *skip* can be encoded as a P frame. The flag *PC2* indicates whether the motion estimation results indicate that motion (e.g., average motion vector magnitude for frame) is larger than a specified threshold. If so, then *PC2* is set to 1.

If there is enough room in the buffer to encode frame *skip* as a P frame (Line A4) and if the estimated motion is large (Line A5), then the current frame *skip* is encoded as a P frame and processing returns to the beginning of the START routine to determine how to encode the next frame in the sequence (Line A6). Otherwise, if there is enough room in the buffer to encode frame *skip* as a P frame (Line A4), but the estimated motion is not large (Lines A5 and A7), then the flag *Pmeet* is set to 1 indicating that there is enough room in the buffer to transmit frame *skip* as a P frame and processing continues to the LOOP1 routine (Line A8). Otherwise, if there is not enough room in the buffer to encode frame *skip* as a P frame (Lines A4 and A10), then the flag *Pmeet* is set to 0 and processing continues to the LOOP2 routine (Line A11). Similarly, if the estimated impact to the buffer based on the raw distortion measure indicates that frame *skip* cannot be transmitted as a P frame (Lines A2 and A13), then the flag *Pmeet* is set to 0 and processing continues to the LOOP2 routine (Line A14).

LOOP1 Routine

As described in the previous section, the LOOP1 routine is called when there is enough room in the buffer to encode the current frame *skip*=*smin* as a P frame, but the motion is not large. Under those circumstances, frame *smin* will be encoded either (1) as a B frame followed by a P frame or (2) in combination with a subsequent frame as a PB frame.

In particular, the LOOP1 routine starts by storing the current frame *smin* as a possible B frame (Line B1 in Fig. 3A). The parameter *skip* is then incremented (Line B2) and the frames from *smin*+1 to *2*smin*-1 are then sequentially checked (Lines B3, B6, B7) to see if any of them can be encoded as a P frame (Lines B4 and B5). This is done by estimating the impact to the buffer and the size of the motion without performing motion estimation (Line B4). If there is enough room in the buffer and the

motion is large, then the frame *skip* is encoded as a P frame and processing returns to the beginning of the START routine for the next frame in the video sequence (Line B5). Note that, when *smin* is 2, only *skip* = 3 is evaluated in this "do while" loop.

If these conditions are not met for any of these frames, then the next frame is selected by setting *skip* equal to $2 * smin$ (Line B8). The number of bits *R* needed to encode frame *skip* as a P frame are then estimated without performing motion estimation and the flag *PBCFD* is set (Line B9). If it is estimated that there is enough room in the buffer to encode frame *smin* and frame *skip* as a PB frame, then *PBCFD* is set to 1. If that condition is satisfied, then motion estimation is performed for frame *skip*, and frames *smin* and *skip*= $2 * smin$ are encoded together as a PB frame (Line B10). Otherwise, there is not enough room to encode those frames as a PB frame, and frame *smin* is encoded as a P frame (Line B11). In either case, processing returns to the START routine (Line B12).

LOOP2 Routine

As described in the section for the START routine, the LOOP2 routine is called when there is not enough room in the buffer to transmit the current frame *skip*=*smin* as a P frame. Under those circumstances, frame *smin* will not be encoded and the LOOP2 routine attempts to select the next frame to be coded and determine how that next frame should be encoded.

In particular, the parameter *skip* is set to *smin*+1 to point to the next frame in the video sequence (Line C1 in Fig. 3B), and the frames from *smin*+1 to *smin*+ $\text{floor}(smin/2)$, where "floor" is a truncation operation, are then sequentially analyzed (Lines C2, C14, C15) to see if any of them can be encoded (Lines C3-C13). For each frame analyzed, the number of bits to encode are calculated based on the raw distortion measure *MAD* and the flags *PCFD1* and *PCFD2* are set to indicate whether there is room in the buffer and whether motion is large, respectively (Line C3). The flag *PCFD2* is set without actually performing motion estimation, by comparing the raw distortion measure *MAD* to a specified threshold level. If *MAD* is greater than the threshold level, then motion is assumed to be large and *PCFD2* is set to 1.

If there is room in the buffer to encode the current frame *skip* as a P frame (Line C4) and if the motion is large (Line C5), then motion estimation is performed and the impact to the buffer (*PC1*) and the motion (*PC2*) are reevaluated using the actual distortion measure *S* (Line C6). If there is still enough room in the buffer (Line C7) and the motion is still large (Line C8), then the current frame *skip* is encoded as a P frame and processing returns to the START routine (Line C8). Otherwise, if the motion-compensated results indicate that there is enough room in the buffer (Line C7), but the actual motion is not large (Line C8 and C9), then the current frame *skip* is stored as a B frame, the pointer *Bframeskip* is set equal to *skip*, the flag *Pmeet* is set to 1 indicating that there is enough room in the buffer to transmit frame *skip* as a P frame, and processing continues to the LOOP3 routine (Line C9).

Otherwise, if the motion-compensated results indicate that there is not enough room in the buffer (Lines C7 and C11), then the current frame *skip* is stored as a B frame, the pointer *Bframeskip* is set equal to *skip*, the flag *Pmeet* is set to 0 indicating that there is not enough room in the buffer to transmit frame *skip* as a P frame, and processing continues to the LOOP3 routine (Line C11). If the non-motion-compensated data indicate that there is enough room in the buffer (Line C4), but the estimated motion is not large (Lines C5 and C13), then the current frame *skip* is stored as a B frame, the pointer *Bframeskip* is set equal to *skip*, the flag *Pmeet* is set to 1 indicating that there is enough room in the buffer to transmit frame *skip* as a P frame, and processing continues to the LOOP3 routine (Line C13). If, however, the non-motion-compensated data indicate that there is not enough room in the buffer (Lines C4 and C14), then processing continues to the next frame (Line C14).

If none of the frames from $sim+1$ to $smin+floor(smin/2)$ satisfies the condition of Line C4, then the flag *Pmeet* is set to 0 indicating that there is not enough room in the buffer to transmit the last frame $skip=smin+floor(smin/2)$ as a P frame, and processing continues to the LOOP3 routine (Line C16).

LOOP3 Routine

As indicated in the previous section, the LOOP3 routine is called when the processing in the LOOP2 routine fails to determine conclusively which frame to encode next and/or how to encode it. In that case, the LOOP3 routine attempts to select the next frame to be coded and determine how that next frame should be encoded.

In particular, the parameter *skip* is set to $smin+floor(smin/2)+1$ (Line D1 in Fig. 3B), and the frames from there up to $2*smin-1$ are then sequentially analyzed (Lines D2, D5, D6) to see if any of them can be encoded (Lines D3-D4). Initializing the parameter *skip* to $smin+floor(smin/2)+1$ allows the P and the B frames to be closer together for the given B skip, which improves coding efficiency in an H.263 PB frame when the P and B frames are tightly coupled. With true B frames, this strategy may need to be changed. For each frame analyzed, the number of bits *R* to encode are calculated based on the estimated distortion measure *Se* generated from the raw distortion measure *MAD* and the flags *PCFD1* and *PCFD2* are set to indicate whether there is room in the buffer and whether motion is large, respectively (Line D3). If both those conditions are met, then the current frame *skip* is encoded as a P frame, and processing returns to the START routine (Line D4).

If the end of the range of frames up to $2*smin-1$ is reached without encoding any of them as a P frame, then *skip* is set equal to the next frame $2*smin$ (Line D7). The number of bits *R* needed to encode frame *skip* as a P frame is then estimated from *MAD* without performing motion estimation and the flag *PBCFD* is set (Line D8). If it is estimated that there is enough room in the buffer to encode the previous frame *Bframeskip* stored as a potential B frame (in LOOP2) and the current frame $skip=2*smin$ as a PB frame (Line D9), then motion estimation is performed for the current frame *skip*

and, if not already performed, for the previous frame stored as a B frame (Line D10). Those frames are then encoded together as a PB frame, and processing returns to the START routine (Line D11).

Otherwise, if those two frames cannot be encoded together as a PB frame (Lines D9 and D12), then, if the previous frame *Bframeskip* stored as a B frame (in LOOP2) can be transmitted as a P frame (i.e., $P_{meet}=1$), then that previous frame *Bframeskip* is encoded as a P frame and processing returns to the START routine (Line D12). Otherwise, if that previous frame cannot be transmitted as a P frame (i.e., $P_{meet}=0$) (Lines D12 and D13), but the non-motion-compensated data indicate that there is room in the buffer (i.e., $PCFD1=1$) and that motion is large (i.e., $PCFD2=1$), then the current frame $skip=2*smin$ is encoded as a P frame, and processing returns to the START routine (Line D13).
 10 Otherwise, processing continues to the LOOP4 routine (Line D14).

LOOP4 Routine

As indicated in the previous section, the LOOP4 routine is called when the processing in the LOOP3 routine fails to determine conclusively which frame to encode next and/or how to encode it. In that case, the LOOP4 routine attempts to select the next frame to be coded and determine how that next frame should be encoded.

In particular, the parameter *skip* is set to $2*smin+1$ (Line E1 in Fig. 3C), and the frames from there up to $smax-1$ are then sequentially analyzed (Lines E2, E6, E7) to see if any of them can be encoded (Lines E3-E5). For each frame analyzed, the number of bits *R* to encode are calculated based on the estimated distortion measure *Se*, which is in turn based on the raw distortion measure *MAD*, and the flag *PBCFD* is set (Line E3). If it is estimated that there is enough room in the buffer to encode the previous frame *Bframeskip* stored as a B frame (in LOOP2) and the current frame *skip* as a PB frame (i.e., $PBCFD=1$), then motion estimation is performed for the current frame *skip* and, if necessary, for the previous frame *Bframeskip* stored as a B frame. Those frames are then encoded together as a PB frame, and processing returns to the START routine (Line E4).

Otherwise, if those two frames cannot be encoded together as a PB frame (i.e., $PBCFD=0$) (Lines E4 and E5) and if the current frame *skip* should be coded as a P frame (i.e., $PCFD1=PCFD2=1$), then the current frame is encoded as a P frame and processing returns to the START routine (Line E5).

30 If the end of the range of frames up to $smax-1$ is reached without encoding any of them, then processing continues to the LOOP5 routine (Line E8).

LOOP5 Routine

As indicated in the previous section, the LOOP5 routine is called when the processing in the LOOP4 routine fails to determine conclusively which frame to encode next and/or how to encode it. In

that case, the LOOP5 routine attempts to select the next frame to be coded and determine how that next frame should be encoded.

In particular, the parameter *skip* is set to $smax+1$ (Line F1 in Fig. 3C), and the frames from there up to $smin + smax$ are then sequentially analyzed (Lines F2, F5, F6) to see if any of them can be encoded (Lines F3-F4). For each frame analyzed, the number of bits R to encode are calculated based on the estimated distortion measure Se , which is in turn based on the raw distortion measure MAD , and the flag $PBCFD$ is set (Line F3). If it is estimated that there is enough room in the buffer to encode the previous frame $Bframeskip$ stored as a B frame (in LOOP2) and the current frame *skip* as a PB frame (i.e., $PBCFD=1$), then motion estimation is performed for the current frame *skip* and, if necessary, for the previous frame $Bframeskip$ stored as a B frame. Those frames are then encoded together as a PB frame, and processing returns to the START routine (Line F4).

If the end of the range of frames up to $smin+smax$ is reached without encoding any of them, then processing continues to the TRANSIENT routine (Line F7).

15 TRANSIENT Routine

As indicated in the previous section, the TRANSIENT routine is called when the processing in the LOOP5 routine fails to determine conclusively which frame to encode next and/or how to encode it. In that case, processing switches from the steady state into the transient state, where the TRANSIENT routine selects one or more frames for encoding as P frames until the TRANSIENT routine determines that processing can return to the steady state. In alternative embodiments, the TRANSIENT routine may encode at least some of the frames as B frames.

In particular, for the current frame *skip*, the raw distortion measure MAD and the number of bits R to encode are calculated based on the estimated distortion measure Se , and the flag $PCFD1$ is set (Line G1). If it is estimated that there is enough room in the buffer to transmit the current frame *skip* as a P frame (i.e., $PCFD1=1$) (Line G2), then motion estimation is performed for the current frame *skip* and the current frame is encoded as a P frame (Line G3). If the buffer occupancy is less than a specified threshold limit $B0$, then processing returns to the steady state of the START routine (Line G4). Otherwise, *skip* is set to $smin$ to select the next frame in the video sequence and processing returns to the start of the TRANSIENT routine to process that next frame (Line G5). If the current frame *skip* could not be transmitted as a P frame (Lines G2 and G7), then *skip* is incremented and processing returns to the start of the TRANSIENT routine to process the next frame, without encoding the current frame (Line G7).

The algorithm presented in Figs. 3A-3C provides a complete approach to frame skipping, PB decision, and quality control when the quantizer step variation is constrained to be within certain bounds from one reference frame to the next. The scheme maintains the user-defined minimum frame

rate during steady-state operations and attempts to transmit data at a high quality and at an "acceptable" frame rate (greater than the minimum frame rate). It provides a graceful degradation in quality and frame rate when there is an increase in motion or complexity. B frames are used both for improving the frame rate and the coded quality. However, in situations of scene change or when the motion increases very rapidly, the demands of frame rate or reference frame quality may be unable to be met. In this situation, processing goes into a transient state to "catch up" and slowly re-enter a new steady state. The scheme requires minimal additional computational complexity and no additional storage (beyond that required to store the incoming frames).

The present invention can be embodied in the form of methods and apparatuses for practicing those methods. The present invention can also be embodied in the form of program code embodied in tangible media, such as floppy diskettes, CD-ROMs, hard drives, or any other machine-readable storage medium, wherein, when the program code is loaded into and executed by a machine, such as a computer, the machine becomes an apparatus for practicing the invention. The present invention can also be embodied in the form of program code, for example, whether stored in a storage medium, loaded into and/or executed by a machine, or transmitted over some transmission medium, such as over electrical wiring or cabling, through fiber optics, or via electromagnetic radiation, wherein, when the program code is loaded into and executed by a machine, such as a computer, the machine becomes an apparatus for practicing the invention. When implemented on a general-purpose processor, the program code segments combine with the processor to provide a unique device that operates analogously to specific logic circuits.

It will be further understood that various changes in the details, materials, and arrangements of the parts which have been described and illustrated in order to explain the nature of this invention may be made by those skilled in the art without departing from the principle and scope of the invention as expressed in the following claims.

CLAIMS

What is claimed is:

- 1 1. A method for processing a sequence of video images, comprising the steps of:
 - 2 (a) generating a raw distortion measure for a current image in the sequence relative to a reference
 - 3 image;
 - 4 (b) using the raw distortion measure to generate an estimate of a motion-compensated distortion
 - 5 measure for the current image relative to the reference image without having to perform motion
 - 6 estimation on the current image;
 - 7 (c) determining whether or how to encode the current image based on the estimate of the motion-
 - 8 compensated distortion measure; and
 - 9 (d) generating a compressed video bitstream for the sequence of video images based on the
 - 10 determination of step (c).
- 1 2. The invention of claim 1, wherein step (a) comprises the steps of:
 - 2 (1) generating a first intermediate distortion measure and a second intermediate distortion measure,
 - 3 wherein:
 - 4 the first intermediate distortion measure characterizes one or more relatively low-distortion
 - 5 portions of the current image; and
 - 6 the second intermediate distortion measure characterizes one or more relatively high-distortion
 - 7 portions of the current image; and
 - 8 (2) generating the raw distortion measure from the first and second intermediate distortion
 - 9 measures.
- 1 3. The invention of claim 2, wherein step (a)(2) applies a correction for double-image effects
- 2 resulting from relative motion between the current image and the reference image.
- 1 4. The invention of claim 1, wherein step (b) comprises the steps of:
 - 2 (1) generating a measure of change in distortion from a previous image in the sequence to the
 - 3 current image; and
 - 4 (2) generating the estimate of the motion-compensated distortion measure based on the measure of
 - 5 change in distortion.
- 1 5. The invention of claim 4, wherein the estimate of the motion-compensated distortion measure
- 2 is generated using a piecewise-linear, continuous function.

1 6. The invention of claim 1, wherein step (c) comprises the steps of:

2 (1) determining whether there is enough room in a corresponding buffer to transmit the current
3 image as a P frame based on the estimate of the motion-compensated distortion measure;

4 (2) determining whether motion in the current image is larger than a specified threshold level
5 based on the raw distortion measure; and

6 (3) determining whether or how to encode the current image based on the results of steps (c)(1)
7 and (c)(2).

1 7. The invention of claim 6, wherein step (c)(1) comprises the step of estimating a number of bits
2 needed to encode the current image as a P frame based on a quadratic rate distortion model.

1 8. The invention of claim 1, wherein step (c) comprises the step of determining whether to (1)
2 skip the current image, (2) encode the current image as a B frame, (3) encode the current image as part
3 of a PB frame, or (4) encode the current image as a reference frame.

1 9. The invention of claim 1, wherein:
2 the processing can be in either a steady state or a transient state;
3 in the steady state, the current image is either skipped or encoded as either a P frame, a B frame, or
4 part of a PB frame; and
5 in the transient state, the current image is either skipped or encoded as either a P frame or a B
6 frame.

1 10. The invention of claim 9, wherein the processing automatically switches from the transient
2 state to the steady state when a corresponding buffer level is below a specified threshold level.

1 / 4

FIG. 1

```

1       $n1 = n2 = 0$ 
2       $dist1 = dist2 = 0$ 
3      For each pixel in current image
4           $ad = \text{absolute difference with corresponding pixel in reference image}$ 
5          If ( $ad < thresh$ )  $dist1 = dist1 + ad, n1 = n1 + 1$ 
6          Else  $dist2 = dist2 + ad, n2 = n2 + 1$ 
7      End for
8       $dist1 = dist1/n1$ 
9       $dist2 = dist2/n2$ 
10      $MAD = (dist1*n1 + dist2*n2*factor + dist1*n2*(1-factor)) / (n1+n2)$ 

```

FIG. 2

```

1       $H = \text{abs}( (MAD(I) - MAD(I-1)) * 2 / (MAD(I) + MAD(I-1)) )$ 
2      If ( $H < T1$ )
3           $Se(I) = S(I-1)$ 
4      ElseIf ( $H < T2$ )
5           $Se(I) = S(I-1) + (k * MAD(I) - S(I-1)) * (H - T1) / (T2 - T1)$ 
6      Else
7           $Se(I) = k * MAD(I)$ 

```

2 / 4

Fig. 3A

START:

```

A1  skip = smin, compute MAD, estimate R(Se), determine PCFD1
A2  If (PCFD1){
A3      Estimate motion, determine S, R(S), PC1, PC2
A4      If (PC1){
A5          If (PC2)
A6              Code frame skip as a P frame, goto START
A7          Else
A8              Pmeet=1, goto LOOP1
A9      }
A10 Else
A11     Pmeet=0, goto LOOP2
A12 }
A13 Else
A14     Pmeet=0, goto LOOP2

```

LOOP1:

```

B1  Store frame smin as a possible B frame
B2  skip = smin + 1,
B3  While (skip < 2*smin)
B4      Compute MAD, estimate R(Se), determine PCFD1, PCFD2
B5      If (PCFD1 && PCFD2) Code frame skip as a P frame, goto START
B6      Else skip = skip + 1,
B7  End while
B8  skip = 2 * smin
B9  Compute MAD, estimate R(Se), rate for B frame smin, determine PBCFD
B10 If (PBCFD) Estimate motion, code frames smin and 2*smin together as a PB frame
B11 Else code frame smin as P frame
B12 goto START

```

3 / 4

Fig. 3B

LOOP2:

```

C1   skip = smin + 1
C2   While (skip <= smin + floor(smin/2))
C3       Compute MAD, estimate R(Se), determine PCFD1, PCFD2
C4       If (PCFD1){
C5           If (PCFD2){
C6               Perform motion estimation, determine S, R(S), PC1, PC2
C7               If (PC1){
C8                   If (PC2) Code frame skip as P frame, goto START
C9                   Else store frame skip as B frame, Bframeskip=skip, Pmeet=1, goto LOOP3
C10              }
C11              Else store frame skip as a B frame, Bframeskip=skip, Pmeet = 0, goto LOOP3
C12          }
C13          Else store frame skip as a B frame, Bframeskip=skip, Pmeet = 1, goto LOOP3
C14      Else skip = skip + 1
C15  End while
C16  Pmeet = 0, goto LOOP3

```

LOOP3:

```

D1   skip = smin + floor(smin/2) + 1
D2   While (skip < 2*smin)
D3       Compute MAD, estimate R(Se), determine PCFD1, PCFD2
D4       If (PCFD1 && PCFD2) Code frame skip as a P frame, goto START
D5       Else skip = skip + 1
D6   End while
D7   skip = 2*smin
D8   Compute MAD, estimate R(Se), rate for B frame Bframeskip, determine PBCFD
D9   If (PBCFD)
D10      Estimate motion
D11      Code frame Bframeskip & frame 2*smin as a PB frame, goto START
D12  Else If (Pmeet = 1) Code frame Bframeskip as a P frame, goto START
D13      Else If (PCFD1 && PCFD2) Code frame 2*smin as a P frame, goto START
D14      Else goto LOOP4

```

Fig. 3C

LOOP4:

```

E1    skip = 2*smin + 1
E2    While (skip < smax)
E3        Compute MAD, estimate R(Se), rate for B frame Bframeskip, determine PBCFD
E4        If (PBCFD) Estimate motion, code frames Bframeskip & skip as PB frame, goto START
E5        Else If (PCFD1 && PCFD2) Code current frame skip as P frame, goto START
E6        Else skip = skip + 1
E7    End while
E8    Goto LOOP5

```

LOOP5:

```

F1    skip = smax + 1
F2    While (skip <= smin + smax)
F3        Compute MAD, estimate R(Se), rate for B frame Bframeskip, determine PBCFD
F4        If (PBCFD) Estimate motion, code frames Bframeskip & skip as PB frame, goto START
F5        Else skip = skip + 1;
F6    End while
F7    Goto TRANSIENT

```

TRANSIENT:

```

G1    Compute MAD, estimate R(Se), determine PCFD1
G2    If (PCFD1){
G3        Estimate motion, code frame skip as P frame
G4        If (buffer occupancy < B0) goto START
G5        Else skip = smin, goto TRANSIENT
G6    }
G7    Else skip = skip + 1, goto TRANSIENT

```


INTERNATIONAL SEARCH REPORT

Inte: International Application No

PCT/US 99/21830

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 H04N7/36 H04N7/26

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 772 362 A (SONY UK LTD)	1
A	7 May 1997 (1997-05-07) column 10, line 6 - line 31 figures 6,7	2-10
P,X	LEE J: "A FAST FRAME TYPE SELECTION TECHNIQUE FOR VERY LOW BIT RATE CODING USING MPEG-1" REAL-TIME IMAGING, GB, ACADEMIC PRESS LIMITED, vol. 5, no. 2, 1 April 1999 (1999-04-01), page 83-94 XP000831435 ISSN: 1077-2014	1,2,6,8
A	page 83, line 1 -page 86, left-hand column, line 30 page 91, right-hand column, line 3 - line 20 --- -/--	3-5,9,10



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"Z" document member of the same patent family

Date of the actual completion of the international search

16 December 1999

Date of mailing of the international search report

22/12/1999

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040. Tx. 31 651 epo nl.
Fax: (+31-70) 340-3016

Authorized officer

Berbain, F

INTERNATIONAL SEARCH REPORT

International Application No.
 PCT/US 99/21830

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	LEE J ET AL: "RATE-DISTORTION OPTIMIZED FRAME TYPE SELECTION FOR MPEG ENCODING" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY,US,IEEE INC. NEW YORK, vol. 7, no. 3, 1 June 1997 (1997-06-01), page 501-510 XP000690588 ISSN: 1051-8215 abstract page 91, right-hand column, line 3 - line 20	1-10
A	TIHAO CHIANG ET AL: "A NEW RATE CONTROL SCHEME USING QUADRATIC RATE DISTORTION MODEL" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY,US,IEEE INC. NEW YORK, vol. 7, no. 1, page 246-250 XP000678897 ISSN: 1051-8215 paragraph '000I! paragraph '00II!	1-10
A	EP 0 514 865 A (MITSUBISHI ELECTRIC CORP) 25 November 1992 (1992-11-25) abstract	1-10

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 99/21830

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0772362 A	07-05-1997	GB 2306831 A	07-05-1997
		JP 9172634 A	30-06-1997
EP 0514865 A	25-11-1992	CA 2068751 A,C	25-11-1992
		DE 69229695 D	09-09-1999
		DE 69229695 T	02-12-1999
		EP 0907292 A	07-04-1999
		JP 5130582 A	25-05-1993
		US 5367335 A	22-11-1994
		US 5402176 A	28-03-1995
		US 5442401 A	15-08-1995

THIS PAGE BLANK (USPTO)